

# VYSOKÉ UCENÍ TECHNICKÉ V BRNE

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MEŘÍCÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SIMULÁTOR ADAPTIVNÍCH REGULAČNÍCH OBVODŮ S BÁZÍ  
PRAVIDEL

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

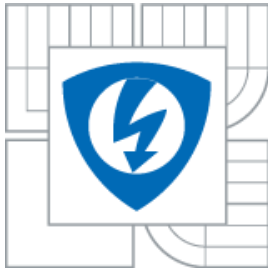
AUTOR PRÁCE  
AUTHOR

MILAN KADLEC

BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**  
ÚSTAV AUTOMATIZACE A MĚŘÍCÍ TECHNIKY

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION**  
**DEPARTMENT OF CONTROL AND INSTRUMENTATION**

# **SIMULÁTOR ADAPTIVNÍCH REGULAČNÍCH OBVODŮ S BÁZÍ PRAVIDEL**

THE SIMULATOR FOR TESTING OF ADAPTIVE CONTROL CIRCUITS

**BAKALÁRSKÁ PRÁCE**  
BACHELOR'S THESIS

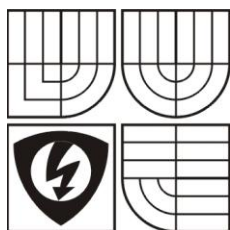
**AUTOR PRÁCE**  
AUTHOR

**MILAN KADLEC**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**prof. Ing. PETR VAVŘÍN, DrSc.**

BRNO 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Milan Kadlec

**Ročník:** 3

**ID:** 109668

**Akademický rok:** 2009/10

**NÁZEV TÉMATU:**

## Simulátor adaptivních regulačních obvodů s bází pravidel

**POKYNY PRO VYPRACOVÁNÍ:**

Vytvořit simulační program pro testování adaptivních regulačních obvodů typu STURE s bází pravidel

**DOPORUČENÁ LITERATURA:**

Manual programu simulink/matlab

Kotek, Kubík a další: Adaptivní a ucíací se systémy, SNTL Praha

**Termín zadání:** 21.9.2009

**Termín odevzdání:** 31.5.2010

**Vedoucí projektu:** prof. Ing. Petr Vavřín, DrSc.

**prof. Ing. Pavel Jura, CSc.**

*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem simulátoru adaptivních regulačních obvodů s bází pravidel. Soustava, která je regulována má druhý řád. Identifikace se provádí metodou nejmenších čtverců. Regulátor je vytvořen, aby kompenzoval kořeny jmenovatele soustavy a tím se řízení značně zjednodušilo.

## **Abstract**

This bachelor's thesis is considered by adaptive control circuit simulator with the base rules. System, which is regulated by a second order. Identification is performed using least squares. The controller is designed to compensate for the roots of the denominator of the system, thereby greatly simplifying management.

## **Klíčová slova**

Regulátor, Adaptivní regulátor, Soustava, Identifikace, Samo se nastavující regulátor

## **Keywords**

Regulator, Adaptive controller, System, Identification, Self tuning regulator

## **Bibliografická citace**

KADLEC, M. *Simulátor adaptivních regulačních obvodů s bází pravidel*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 45 s. Vedoucí bakalářské práce prof. Ing. Petr Vavřín, DrSc..

## **Prohlášení**

Prohlašuji, že svou bakalářskou práci na téma Simulátor adaptivních regulačních obvodů s bázi pravidel jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 26. 5. 2010

Podpis autora .....

## **Poděkování**

Děkuji vedoucímu bakalářské práce Prof. Ing. Petru Vavřínovi, DrSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 26. 5. 2010

Podpis autora .....

## Obsah

|  |           |
|--|-----------|
| <b>1. ÚVOD .....</b>                               | <b>11</b> |
| <b>2. PROGRAMY .....</b>                           | <b>12</b> |
| 2.1 Matlab .....                                   | 12        |
| 2.2 Simulink .....                                 | 13        |
| 2.2.1 Blocksety .....                              | 15        |
| 2.2.2 Model-Based design .....                     | 15        |
| <b>3. ADAPTIVNÍ REGULACE.....</b>                  | <b>16</b> |
| 3.1 Úvod.....                                      | 16        |
| 3.2 Samo se nastavující regulátor (STR) .....      | 17        |
| 3.3 Problémy adaptivního řízení .....              | 19        |
| 3.4 Aplikace .....                                 | 19        |
| 3.4.1 Automatické ladění.....                      | 19        |
| <b>4. IDENTIFIKACE .....</b>                       | <b>21</b> |
| 4.1 Identifikace procesu pro adaptivní řízení..... | 23        |
| <b>5. REGULÁTORY .....</b>                         | <b>26</b> |
| 5.1 PID regulátor.....                             | 26        |
| 5.2 PSD regulátor.....                             | 29        |
| <b>6. ŘEŠENÍ OBVODU.....</b>                       | <b>30</b> |
| 6.1 Zadání .....                                   | 30        |
| 6.2 Báze pravidel .....                            | 30        |
| 6.3 Identifikace .....                             | 35        |
| 6.4 Přepoččet regulátoru .....                     | 37        |
| 6.5 Realizace PSD regulátoru .....                 | 39        |
| 6.6 Příklady .....                                 | 41        |
| <b>7. ZÁVĚR.....</b>                               | <b>44</b> |
| <b>LITERATURA .....</b>                            | <b>45</b> |



## SEZNAM OBRÁZKŮ:

|   |    |
|---|----|
| Obrázek 1: Program MATLAB .....   | 13 |
| Obrázek 2: Program SIMULINK.....  | 14 |
| Obrázek 3: Knihovna programu SIMULINK .....   | 15 |
| Obrázek 4: Blokové schéma adaptivní regulace .....  | 17 |
| Obrázek 5: Blokové schéma samo se nastavujícího regulátoru (STR).....                     | 18 |
| Obrázek 6: Spojitý PID regulátor.....   | 28 |
| Obrázek 7: Stavový diagram polohového PSD regulátoru .....                                | 29 |
| Obrázek 8: Teoretické blokové schéma adaptivní regulace s diskretním regulátorem<br>..... | 30 |
| Obrázek 9: Blok ARX z knihovny SIMULINK .....   | 35 |
| Obrázek 10: Blokové schéma reálného procesu .....   | 41 |
| Obrázek 11: Přejchodová charakteristika pro příklad 1 .....                               | 42 |
| Obrázek 12: Přejchodová charakteristika pro příklad 2 .....                               | 43 |

## SEZNAM SYMBOLŮ A ZKRATEK

|                 |  |
|-----------------|--|
| STR             | Samo se nastavující regulátor.   |
| PID             | Spojité regulátor obsahující složky: zesílení, integraci, derivaci.            |
| PSD             | Diskrétní regulátor obsahující složky: zesílení, sumaci, diferenci.            |
| $k_r, T_1, T_2$ | Konstanty PID regulátoru s konstantami v čitateli.                             |
| $K_R, T_I, T_D$ | Konstanty PID regulátoru s vyjádřením integrační a derivační časové konstanty. |
| $\varepsilon$   | Realizační konstanta u regulátorů obsahující derivaci.                         |
| $K, d_s, d_d$   | Konstanty PSD regulátoru   |
| ARX             | Metoda identifikace, metoda nejmenších čtverců                                 |
| $T_s$           | Perioda vzorkování   |

## 1. ÚVOD

Úkolem této bakalářské práce je navrhnout simulační program pro testování adaptivních regulačních obvodů typu STURE s bází pravidel. Práce bude tvořena v prostředí programu MATLAB a jeho nadstavby SIMULINK. Práce spočívá v tom na libovolnou soustavu, která je v daném tvaru, navrhnout regulátor pomocí pravidel, které jsou zásadní pro realizaci.

Pokud bude úloha řešena v bodech, tam vyplyne tato posloupnost dějů v obvodu:

- natavení soustavy
- identifikace soustavy v průběhu procesu
- výpočet parametrů regulátoru
- realizace regulátoru

Regulace může být kvalitní pouze tehdy, když je regulátor vhodně nastaven pro zadanou soustavu. V praxi se však vlastnosti soustavy mohou v průběhu regulace měnit. Je proto zapotřebí měnit odpovídajícím způsobem parametry regulátoru. Vhodným řešením je tzv. Adaptivní regulátor, který automaticky vyhodnocuje průběh regulačního pochodu a podle toho upravuje hodnoty parametrů regulátoru.

## 2. PROGRAMY

Pro tvorbu programu bakalářské práce byl zadán program MATLAB a jeho nadstavba SIMULINK.

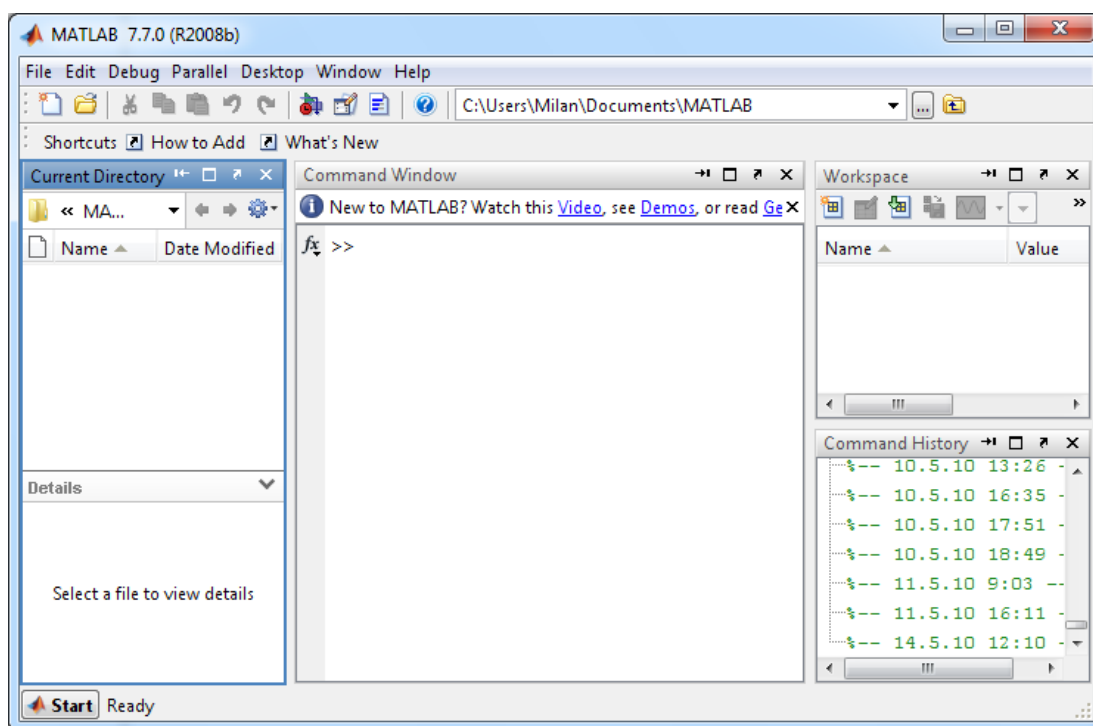
### 2.1 MATLAB

MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. MATLAB je nástroj jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací.

Výpočetní systém MATLAB se během uplynulých let stal celosvětovým standardem v oblasti technických výpočtů a simulací ve sféře vědy, výzkumu, průmyslu i v oblasti vzdělávání.

MATLAB poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti.

Díky své architektuře je MATLAB určen zejména těm, kteří potřebují řešit početně náročné úlohy a přitom nechtějí nebo nemají čas zkoumat matematickou podstatu problémů. Více než milion uživatelů po celém světě využívá možnosti jazyka MATLABu, který je mnohem jednodušší než například Fortran nebo C a který skýtá obrovský potenciál produktivity a tvořivosti. Za nejsilnější stránku MATLABu je považováno mimořádně rychlé výpočetní jádro s optimálními algoritmy, které jsou prověřeny léty provozu na špičkových pracovištích po celém světě. MATLAB byl implementován na všech významných platformách (Windows, Linux, Solaris, Mac).[ 1 ]



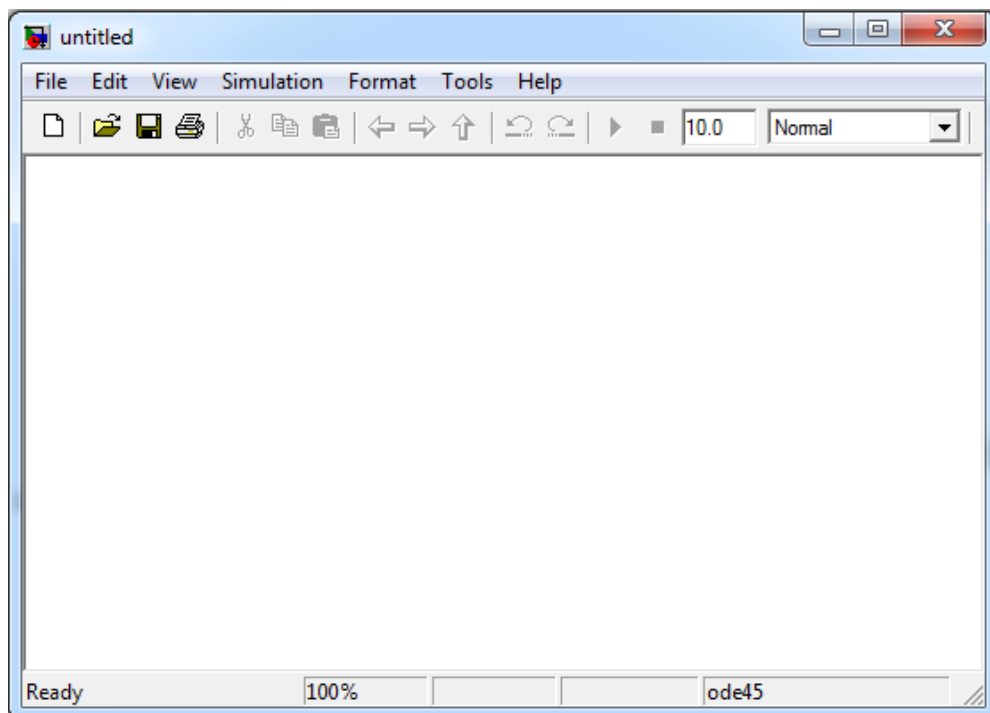
Obrázek 1: Program MATLAB

## 2.2 SIMULINK

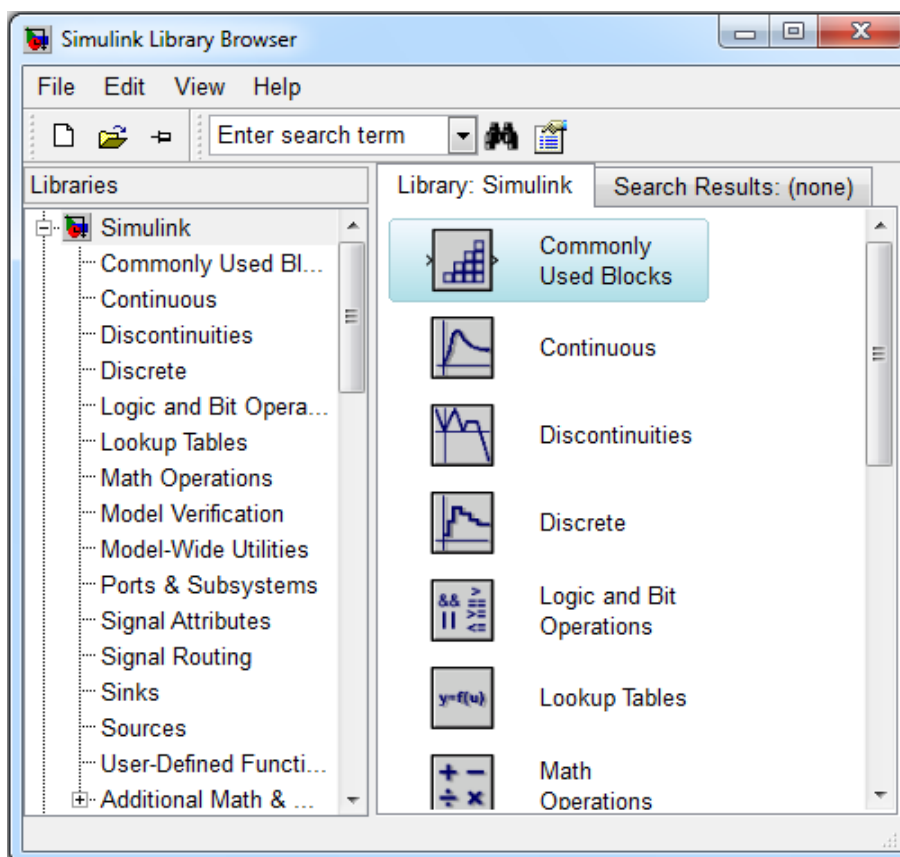
Simulink je nadstavba MATLABu pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení nelineárních diferenciálních rovnic. Poskytuje uživateli možnost rychle a snadno vytvářet modely dynamických soustav ve formě blokových schémat a rovnic.

Kromě standardních úloh dovolu je Simulink rychle a přesně simulovat i rozsáhlé "stiff" systémy s efektivním využitím paměti počítače. Pomocí Simulinku a jeho grafického editoru lze vytvářet modely lineárních, nelineárních, v čase diskretních nebo spojitých systémů pouhým přesouváním funkčních bloků myši. Simulink také umožňuje spouštět určité části simulačního schématu na základě výsledku logické podmínky. Tyto spouštěné a povolované subsystemy umožňují použití programu v náročných simulačních experimentech. Samozřejmostí je otevřená architektura, která dovolu je uživateli vytvářet si vlastní funkční bloky a rozšiřovat již tak bohatou knihovnu Simulinku. Hierarchická struktura modelů umožňuje koncipovat i velmi

složité systémy do přehledné soustavy subsystémů prakticky bez omezení počtu bloků. Simulink, stejně jako MATLAB, dovoluje připojovat funkce napsané uživateli v jazyce C. Vynikající grafické možnosti Simulinku je možné přímo využít k tvorbě dokumentace. Mezi neocenitelné vlastnosti Simulinku patří nezávislost uživatelského rozhraní na počítačové platformě. Přenositelnost modelů a schémat mezi různými typy počítačů umožňuje vytvářet rozsáhlé modely, které vyžadují spolupráci většího kolektivu řešitelů na různých úrovních. [ 1 ]



**Obrázek 2: Program SIMULINK**



**Obrázek 3: Knihovna programu SIMULINK**

### 2.2.1 Blocksety

Otevřená architektura Simulinku vedla ke vzniku knihoven bloků, nazývaných blocksety, které rozšiřují základní knihovnu bloků Simulinku a umožňují použití programu v příslušných vědních a technických oborech. Knihovny je možné rozšiřovat i o vlastní bloky, vytvořené uživatelem. [ 1 ]

### 2.2.2 Model-Based design

Model Based Design (MBD) je metoda rychlého a efektivního návrhu dynamického systému. Pomocí MBD lze navrhovat řídicí systémy, systémy pro zpracování signálu a obrazu i komunikační systémy. [ 1 ]

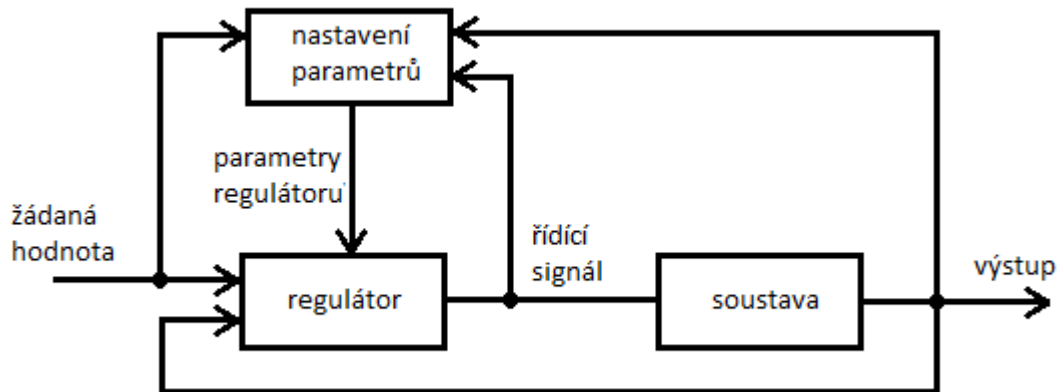
### 3. ADAPTIVNÍ REGULACE

#### 3.1 ÚVOD

V běžném jazyce "adaptivní", znamená změnu chování v souladu k novým okolnostem. Intuitivně, adaptivní regulátor je tedy správcem, že může změnit své chování v reakci na změny v dynamice procesu a charakter poruchy. Vzhledem k tomu, běžné zpětné vazby se také pokouší snížit dopady poruch a růst nespolehlivosti, otázka jaký je rozdíl mezi zpětnou vazbou a adaptivním řízením. V průběhu let došlo k mnoha pokusům definovat oficiálně adaptivní řízení. Na počátku symposia v roce 1961 po dlouhé diskusi skončila s tímto návrhem: „Adaptivní systém je každý fyzikální systém, který byl navržen s adaptivního hlediska.“ Obnovený pokus byl vyroben IEEE výbor v roce 1973. Navrhla novou slovní zásobu na základě pojmů jako samostatně organizovat kontrolu (SOC) systému, adaptivní parametry (SOC), adaptivní výkon (SOC), a učení kontrolní systém. Avšak tyto snahy nebyly široce přijímané. Smysluplné vymezení adaptivní řízení, které by bylo možné podívat se na správce hardwaru a softwaru a rozhodnout, zda je či není adaptivní stále chybí. Nicméně se zdá, že shoda, že neustálý-přírůstek systému zpětné vazby není adaptivní systém.

Adaptivní regulátor je regulátor s nastavitelnou hodnotou parametrů a mechanismus pro nastavení parametrů. Regulátor se stane nelineární, z důvodu parametru opravného mechanismu. Je však velmi zvláštní konstrukce. Vzhledem k tomu, že obecně u nelineárních systémů je obtížné se s tím vypořádat dává smysl, aby zvažila speciální třídy nelineárních systémů. Adaptivní systém řízení může být myšlenka, jak mít dvě smyčky. Jedna smyčka je normální zpětné vazby s procesem a regulátorem. Další smyčka je na úpravu parametrů smyčky. Nastavení parametrů je často pomalejší než normální zpětná vazba. Řídící inženýr by měl vědět o adaptivních systémech jejich vhodné vlastnosti, které mohou být ziskové. Slouží pro návrh řídicího systému pro vyšší výkon a funkčnost.[ 2 ]





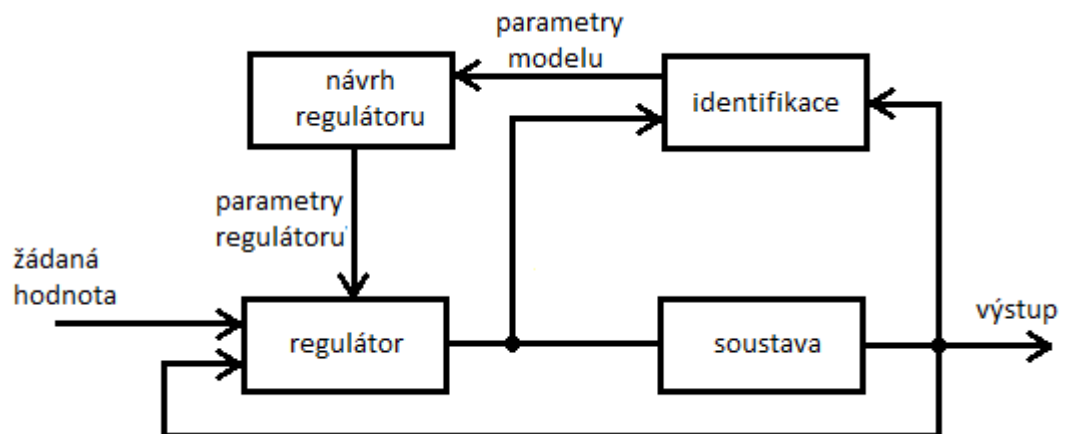
Obrázek 4: Blokové schéma adaptivní regulace

### 3.2 SAMO SE NASTAVUJÍCÍ REGULÁTOR (STR)

Tento systém samo se nastavujících regulátorů, je získán v případě, že odhady parametrů procesu jsou aktualizovány a správce parametry jsou získány z řešení návrhu systému pomocí odhadů parametrů. Blokové schéma takového systému je uvedeno na Obrázku 5. Myšlenka adaptivního regulátoru může být jako složení ze dvou smyček. Vnitřní smyčka se skládá z procesu a obyčejnou zpětnou vazbou a regulátoru. Parametry regulátoru jsou upraveny vnějším vedením, které se skládá z odhadu rekurzivních parametrů a statických výpočtů. Někdy je možné odhadnutí parametrů procesu, aniž by zkoumal řídicí signály. Všimněte si, že systém může být chápán jako automatizace procesů modelování a návrhu, ve kterém jsou procesního modelu a návrhu řízení jsou aktualizovány při každé periodě odběru vzorku. Správce této konstrukce se nazývá self-tuning regulator (STR) zdůraznit, že správce automaticky naladí její parametry pro získání požadované vlastnosti systém uzavřené smyčky.

Blok označený návrh regulátoru představuje on-line řešení návrhu pro systém se známými parametry. To je základní návrhový problém. Takový problém může v souvislosti s většinou adaptivních systémů kontroly, ale to je často dávana nepřímo.

Chcete-li hodnotit adaptivitu, tak musí být vlastnosti systému za ideálních podmínek, kdy jsou parametry přesně známy.



**Obrázek 5: Blokové schéma samo se nastavujícího regulátoru (STR)**

Tento režim STR je velmi flexibilní, pokud jde o výběr základního návrhu a metody odhadu. Mnoho různých kombinací byly prozkoumány. Parametry regulátoru jsou aktualizovány nepřímo přes konstrukční výpočty v samostatném bloku návrh regulátoru. To je někdy možné přeparametrizování procesu tak, aby se pomocí modelu mohl vyjádřit regulátor. To dává významné zjednodušení algoritmu, protože konstrukční výpočty jsou eliminovány. Pokud jde o Obrázek 5, tak zmizí blok návrhu regulátoru a parametry jsou aktualizovány přímo.

V STR identifikace nebo parametry procesu jsou odhadnuty v reálném čase. Odhady jsou pak používány jako by se rovnali skutečným parametrům. Toto je nazýváno jistota princip rovnocennosti. V mnoha identifikačních režimech je také možné se dostat z měřítek kvality odhadů. Tato nejistota pak může být použita při návrhu regulátoru. Například v případě, že je velká nejistota jeden může zvolit konzervativní návrh. [ 2 ]

### 3.3 PROBLÉMY ADAPTIVNÍHO ŘÍZENÍ

Adaptivní regulátor byl definován jako regulátor s nastavitelnými parametry a mechanismus pro nastavení parametrů. Stavba adaptivního regulátoru tedy obsahuje následující kroky:

- Charakterizují požadované chování uzavřené smyčky systému.
- Stanovit vhodné kontrolní pravidla na nastavení parametrů.
- Nalézt mechanismus pro nastavení parametrů.
- Implementovat zákon o kontrole.

### 3.4 APLIKACE

Bylo množství žádostí adaptivního řízení zpětné vazby od poloviny 1950. Rané pokusy, které používají analogové implementace byli sužovány problémy s hardwarem. Systémy implementovány pomocí minipočítače se objevil na začátku 1970. Počet žádostí vzrostl drasticky s příchodem mikroprocesoru, který učinil technologii nákladově efektivní. Adaptivní techniky bylo použito v pravidelných průmyslových regulátorech od počátku 1980. Dnes velký počet průmyslových řídicích smyček je pod adaptivním řízením. Ty zahrnují širokou škálu aplikací v oblasti letectví, řízení procesů, řízení lodí, robotiky, automobilového průmyslu a biomedicínských systémů. Aplikace ukázaly, že existuje mnoho případů, kdy adaptivní řízení je velmi užitečné, jiné, které výhody jsou okrajové a ještě další, v němž jsou nevhodné. Na základě výrobků a jejich použití je zřejmé, že adaptivní techniky může být použito v mnoha různých způsobech. [ 2 ]

#### 3.4.1 Automatické ladění

Nejrozšířenější aplikací jsou v automatickém ladění regulátorů. Tím, automatické ladění, máme na mysli, že parametry standardního regulátoru, například regulátor PID, jsou naladěné automaticky při požadavku provozovatele. Po doladění

parametrů udržují konstantní hodnotu. Prakticky všechny řadiče mohou využívat nástroje pro automatické ladění. Tím se výrazně zjednodušilo použití regulátorů. Prakticky všechny adaptivní techniky mohou být použity pro tento účel. Jedinou smyčku regulátorů a distribuovaných systémů pro řízení procesů jsou významné aplikační oblasti. Většina z těchto regulátorů jsou typu PID. To je obrovská oblast použití, protože tam jsou miliony regulátorů tohoto typu v provozu. Mnohé z nich jsou špatně naladěné.

I když automatické ladění je v současnosti široce používá v jednoduchých regulátorech, to je také prospěšná pro složitější regulátory. To je v podstatě podmínkou pro široké využití pokročilých řídicích algoritmů. Mechanismus pro automatické ladění je často nutné získat ve správném časovém rozsahu a najít výchozí hodnotu pro složitější adaptivní regulátory. Hlavní výhodou použití automatického tunery byly také vyvinuty pro ostatní standardní aplikace, jako je řízení motoru. To je také případ, ve kterém poměrně standardizovaný systém musí být aplikován na širokou škálu aplikací. [ 2 ]

## 4. IDENTIFIKACE

Pro návrh regulátoru potřebujeme znát matematický model regulované soustavy. Existuje sice několik postupů, které umožňují navrhnout algoritmus řízení bez popisu vlastností soustavy, používáme je však spíše jako krajní řešení v těch případech, kdy formulace matematického modelu je buď nemožná, nebo velmi obtížná. Pro běžné a linearizovatelné soustavy však není obvykle příliš obtížné najít alespoň přibližný popis-model. Lze k němu dospět buď analyticky, tj. formulací příslušných diferenciálních či diferenčních rovnic (na základě fyzikálně chemických dějů, které v soustavě probíhají) nebo experimentálně, měřením statických i dynamických vlastností reálného objektu.

Při tomto způsobu identifikace obvykle používáme pro buzení soustavy nějaký typický signál. Nejčastěji je to skoková změna, nebo harmonický průběh. Třetí standardní časový průběh, totiž jednotkový impuls, je obtížně realizovat a používá se spíše výjimečně. Použijeme-li skok vstupní veličiny, obdržíme jako odezvu přechodovou charakteristiku. To ovšem platí za předpokladu nulových počátečních podmínek a při absenci poruchových signálů (na soustavu kromě vstupní veličiny nepůsobí žádný jiný signál). Budeme-li soustavu budít harmonickým signálem, jehož frekvenci budeme postupně měnit, můžeme měřit zesílení a fázový posun procházejícího signálu a získat tak jednotlivé body frekvenční charakteristiky. S ohledem na praktické podmínky lze doporučit:

- Měření přechodové charakteristiky je vhodné pro soustavy s předpokládanými časovými konstantami v rozmezí jednotek až tisíců sekund; zapisovače pro takové časové průběhy jsou běžně dostupné a realizace dostatečně věrné skokové změny vstupní veličiny je možná. Vzorkování je mikroprocesorem.
- Měření s použitím harmonického signálu je vhodné spíše pro rychlejší soustavy, neboť po každé změně frekvence je třeba počkat, až dozní přechodový děj vyvolaný touto změnou. Totéž platí v případě, kdy nejsou zaručeny nulové počáteční podmínky. Nevýhodou

frekvenčního měření je nutnost předem odhadnout frekvenční rozsah, ve kterém se dynamické vlastnosti soustavy projeví.

Společnou nevýhodou měření přechodové charakteristiky nebo jednotlivých bodů frekvenční charakteristiky je nutnost izolovat soustavu od jiných signálových vlivů. To je možné jen při vyřazení soustavy z běžného provozu. Existuje i řada tzv. “on-line“ postupů, které určují potřebný matematický model na základě dlouhodobého měření vstupních a výstupních hodnot. Nejpoužívanější je metoda minima součtu kvadrátů odchylek. Její princip ukážeme na jednoduchém příkladě.

Předpokládejme, že identifikovaná soustava má předpokládaný diskrétní přenos ve tvaru:

$$F(z) = \frac{a}{z - b} \quad (1)$$

Diskrétní přenos používáme pro jednoduchost vysvětlení. Úkolem identifikace je určit hodnoty parametrů  $a$ ,  $b$ . Vstupní veličina je  $x(k)$ , výstupní je  $y(k)$ , kde  $k$  je krok diskrétního signálu. Z přenosu plyne, že platí následující rovnice :

$y(k+1) = ax(k) + by(k)$ . Teoreticky by tedy stačilo ( za předpokladu nulových počátečních podmínek a absence vlivu působení poruchového signálu) změřit dvě sobě odpovídající dvojice vstupních a výstupních hodnot. Tím získáme dvě rovnice o dvou neznámých ( $a$ ,  $b$ ), které lze řešit za předpokladu, že matice soustavy rovnic není singulární. To bude splněno, jestliže hodnoty vstupního signálu budou různé. Pokud provedeme větší množství měření, než je nutné pro výpočet neznámých parametrů soustavy, získáme možnost zmenšit vliv nenulových počátečních podmínek i případného poruchového signálu, který si můžeme představit jako chybu prováděných měření. Je známo, že tento postup, že tento postup bude úspěšný, pokud poruchový signál bude mít určité statické parametry (např. nulovou střední hodnotu). Nevýhodou této metody je nutnost předem určit tvar matematického modelu (počet neznámých parametrů v čitateli i jmenovateli přenosu). Podobné metody existují i pro stanovení spojitého přenosu. [ 3 ]

#### 4.1 IDENTIFIKACE PROCESU PRO ADAPTIVNÍ ŘÍZENÍ

V adaptivním řízení je úloha identifikace právě tak důležitá jako role syntézy regulátoru. Identifikace pro adaptivní řízení má ovšem svá specifika, která vedou k tomu, že se v převážné míře odhadují parametry regresního modelu (ARX) a používá se metoda nejmenších čtverců. Identifikace pro adaptivní řízení vychází z následujících podmínek:

- Vstupy do řízeného procesu jsou generovány regulátorem.
- V řídicí smyčce se vyskytují poruchy, které by měl regulátor kompenzovat a stabilizovat proces. Přítomnost poruch zhoršuje možnosti identifikace parametrů a regulátor z těchto odhadů může být špatně určen.
- Identifikační proces u adaptivního řízení trvá velmi dlouho (po dobu trvání řízení procesu). Proto lze jen stěží předpokládat, že odhadované parametry budou konstantní.
- Identifikace musí dávat výsledky za různých pracovních podmínek procesu (např. 30%, 70%, 100% požadovaného výkonu, a to i v období relativního stacionárního stavu, při poruchách nebo přechodech mezi různými stavy).
- Strukturu identifikovaného modelu obvykle neměníme.
- Identifikační algoritmus musí být numericky spolehlivý a dostatečně rychlý.
- Poměrně důležitou roli hraje počáteční nastavení parametrů regulátoru především proto, aby se zabránilo nekorektním akčním zásahům na počátku identifikace. Odhady parametrů by měly tedy buď již na počátku identifikace dostatečně reprezentovat daný proces, nebo regulátor má na počátku identifikace přednastaveny parametry tak, že regulační pochod je pro daný proces přijatelný (např. s přetlumenou dynamikou), nestačí aby byl stabilní.

- Počátečními podmínkami nejběžněji používané identifikační metody jsou počáteční odhady parametrů a jejich kovarianční matice. Zatímco role počátečních odhadů parametrů je respektována, role kovarianční matice bývá nedoceněna a její návrh je obtížný. Ukazuje se, že schůdnou a poměrně jednoduchou metodou, jak získat počáteční podmínky pro identifikaci, zahrnující v podstatě libovolnou apriorní informaci, je metoda fiktivních dat. Její podstata spočívá v tom, že pomocí velmi zjednodušeného diskrétního modelu procesu, který dostatečně reprezentuje proces, jsou vygenerována data.
- Zpracováním těchto fiktivních dat podobně jako by byly získány z reálného procesu, lze získat počáteční odhady a kovarianční matici. Problém je, že tato data nelze zpracovat obvyklým postupem při použití metody nejmenších čtverců. Je si třeba uvědomit, že jednotlivé dílčí složky apriorní informace mohou být i částečně protichůdné, ale v každém případě je třeba tuto informaci brát jen s určitou pravděpodobností. Přitom by se mohlo stát, že použití velkého počtu dat o specifické informaci (např. zesílení) při nepřesném modelu by vedlo k tomu, že by se tato informace v odhadech tak zafixovala, že ani velké množství skutečných dat by ji již nezměnilo. Proto tato data použijeme pouze jednorázově pro návrh regulátoru. Znovu nastavíme počáteční podmínky pro identifikaci a jistý čas regulujeme pouze s takto nastaveným regulátorem a přitom stále identifikujeme proces. Po shromáždění dostatečného množství dat můžeme přejít k adaptivnímu řízení a přeepsání parametrů řídicího algoritmu.
- Identifikace procesů při časově proměnných parametrech je možno řešit technikou zapomínání. Nejznámější je exponenciální zapomínání, kde vliv starších dat na odhady parametrů a jejich kovarianční matici exponenciálně klesá. Závažným nedostatkem tohoto zapomínání v adaptivním režimu je ztráta informace v případech, kdy je proces natolik ustálený, že data přináší jen málo informace o vlastnostech procesu. Tuto situaci je třeba řešit



vypínáním identifikace, proměnným koeficientem zapomínání nebo jinými formami zapomínání, které v sobě obsahují schopnost měnit množství zapomínané informace podle charakteru dat. Ve své podstatě parametry procesu jsou nejlépe určeny, když na proces působí časté změny žádané hodnoty, je potřebná pestrá změna vstupních veličin. V ustáleném stavu po dosažení žádané hodnoty je lépe proces identifikace změnou koeficientu zapomínání pozastavit a tím zajistit, aby dynamika poruch se nedostávala do identifikovaných parametrů. Při další změně žádané hodnoty, která přesáhne nastavenou hranici, potom opět obnovíme proces identifikace. [ 4 ]

## 5. REGULÁTORY

Regulátor je zařízení pro ovlivňování regulovaného systému, automatizovanou regulaci, k dosažení a udržení jeho požadovaného stavu. Typicky se používá v záporné zpětné vazbě systému. Vstupem regulátoru pak nebývá přímo sledovaná veličina jako výstup celého systému, ale jen odchylka od požadované hodnoty. Regulátor pak reguluje systém s cílem buď úplné eliminace odchylky, nebo jeho regulační zásahy odchylku alespoň udržují v předepsaných mezích. Regulátor čte stavy systému, a to buď přímo, anebo, jsou-li nedosažitelné, si je rekonstruuje vlastním modelem. Modelováním systémů a jejich regulátorů se zabývá teorie řízení. Regulátor bývá na systém připojen přes vstupní a výstupní převodníky. Regulace je při čtení systému v čase buď spojitá, nebo diskrétně vzorkovaná. I zásahy do systému mohou být buď analogové, nebo digitální, stupňovité. [ 5 ]

PID regulátory jsou zdaleka nejrozšířenější regulátory používané v technické praxi. V současnosti se až na výjimky spojitě regulátory realizované operačními zesilovači nevyskytují a místo nich jsou používány diskrétní regulátory PSD, jejichž algoritmus pouze napodobuje chování spojitěho členu.

Proces vzorkování a následného tvarování D/A převodníkem negativně ovlivňuje stabilitu zpětnovazebního regulačního obvodu. Existence číslicového regulátoru pracujícího se vzorkovací periodou  $T_S$  se projeví na stabilitě obvodu jako dopravní zpoždění s dobou  $\tau = T_S / 2$ . Proto i kritické zesílení a kritická perioda, obvodu, v němž je použit číslicový regulátor, závisí na periodě vzorkování, se kterou regulátor pracuje. [ 4 ]

### 5.1 PID REGULÁTOR

Časový popis chování PID regulátoru je popsán následující rovnicí:

$$u(t) = K \left( e(t) + \frac{1}{T_I} \int_0^t e(t) + T_D \frac{de(t)}{dt} \right) \quad (2)$$

akční zásah  $u(t)$  je tedy počítán jako součet tří složek: regulační odchylka  $e(t)$  – proporcionální složka – udávající rozdíl mezi požadovaným výstupem soustavy a jejím skutečným stavem, integrací regulační odchylky – integrační složka – a její časovou derivací – derivační složka.

Vliv jednotlivých složek je dán příslušnými konstantami:

K- proporcionální konstanta regulátoru

$T_I$ - integrační konstanta regulátoru

$T_D$ - derivační konstanta regulátoru

Proporcionální složka zabezpečuje základní chování zpětnovazebního regulačního členu – čím více se liší výstup soustavy od požadované hodnoty, tím větší je akční zásah, který má tuto chybu napravit.

Pro odstranění trvalé regulační odchylky slouží integrační složka. Existuje-li nenulová regulační odchylka, přičítá integrační složka trvale k akčnímu zásahu jí úměrnou hodnotu, dokud odchylka nevymizí. Na druhou stranu integrační složka snižuje stabilitu zpětnovazebního obvodu.

Ke zvýšení stability a tím i zrychlení regulačního děje slouží derivační složka. Derivační složka slouží jako „brzda“ – čím rychleji se regulační odchylka mění, tím více působí derivační složka proti směru změny. [ 4 ]

Operátorový přenos PID regulátoru je popsán následující rovnicí:

$$F_R(p) = r_0 + \frac{r_i}{p} + r_d p = K_R \left( 1 + \frac{1}{T_I p} + T_D p \right) = k_r \frac{(T_1 p + 1)(T_2 p + 1)}{p} \quad (3)$$

Mezi konstantami pro různé formy přenosu regulátoru platí tyto vztahy:

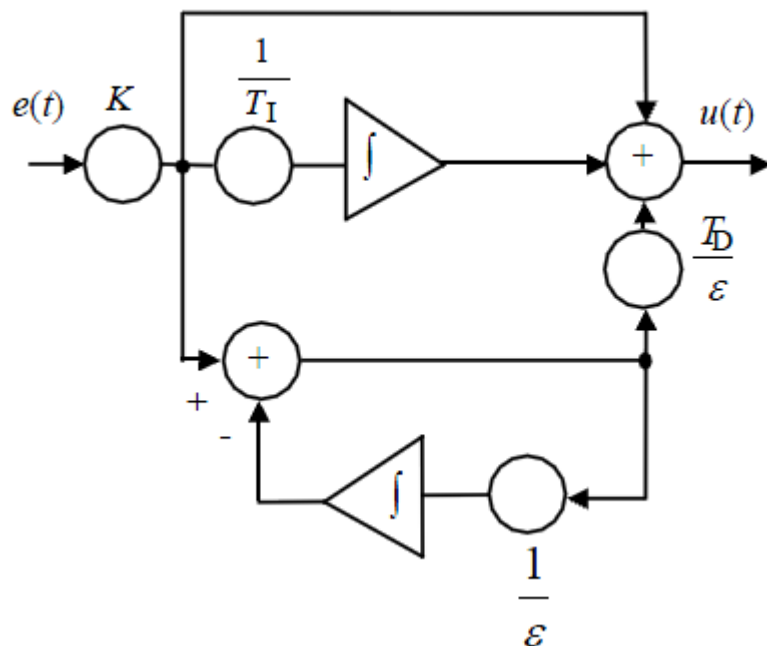
$$K_R = r_0 \quad T_D = \frac{r_d}{r_0} \quad T_I = \frac{r_0}{r_i} \quad k_r = r_i \quad T_{1,2} = \frac{-T_I \pm \sqrt{T_I(T_I - 4T_D)}}{2T_I T_D} \quad (4)$$

Velmi častý je druhý tvar, kde jsou zřejmé jednotlivé konstanty regulátoru, v praxi často používané. Tyto konstanty byly již výše popsány ( $K_R$ ,  $T_I$ ,  $T_D$ ). Tento tvar je též vhodný pro praktickou realizaci PID regulátoru, protože umožňuje vypořádat se s problémy, které přináší omezení akčního zásahu. Poslední tvar regulátoru není zcela rovnocenný předchozím regulátorům, neboť předpokládá pouze reálné časové konstanty.

Zde popsané přenosy regulátoru, však nejsou realizovatelné z důvodu vyššího řádu čitatele, než jmenovatele. Z tohoto důvodu se do přenosů přidává realizační konstanta  $\varepsilon$ , která bývá o dva řády menší než nejmenší časová konstanta. Realizační konstanta je volena tedy tak, aby co nejméně ovlivňovala chování ideálních regulátorů. [ 3 ]

Upravené přenosy regulátoru:

$$F_R(p) = K_R \left( 1 + \frac{1}{T_I p} + \frac{T_D p}{\varepsilon p + 1} \right) = k_r \frac{(T_1 p + 1)(T_2 p + 1)}{p(\varepsilon p + 1)} \quad (5)$$



Obrázek 6: Spojitý PID regulátor

## 5.2 PSD REGULÁTOR

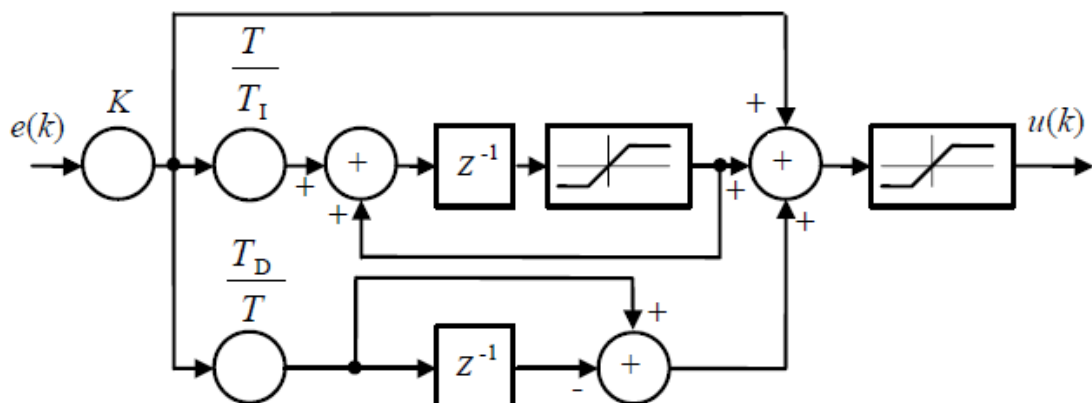
Nejjednodušší a nejrozšířenější tvar PSD regulátoru získáme náhradou spojitě integrační složky za numerickou integraci – sumaci a spojitou časovou derivaci nahradíme dvoubodovou diferencí. Protože při numerické integraci počítáme plochu pod průběhem regulační odchylky jako součet obdélníků, násobíme sumaci periodou vzorkování. Při výpočtu numerické derivace (rychlost změny regulační odchylky) musíme vzorkovací periodou dělit. Výsledná rovnice udávající výpočet akčního zásahu PSD regulátoru má tedy tvar:

$$u(k) = K \left( e(k) + \frac{T_s}{T_I} \sum_{i=1}^k e(i) + \frac{T_D}{T_s} (e(k) - e(k-1)) \right) \quad (6)$$

Kde  $T_s$  je perioda vzorkování a  $k$  je diskretní krok. Tento typ regulátoru se nazývá polohový (paralelní) PSD regulátor.

Tento vztah můžeme převést na diskretní přenosovou funkci PSD regulátoru:

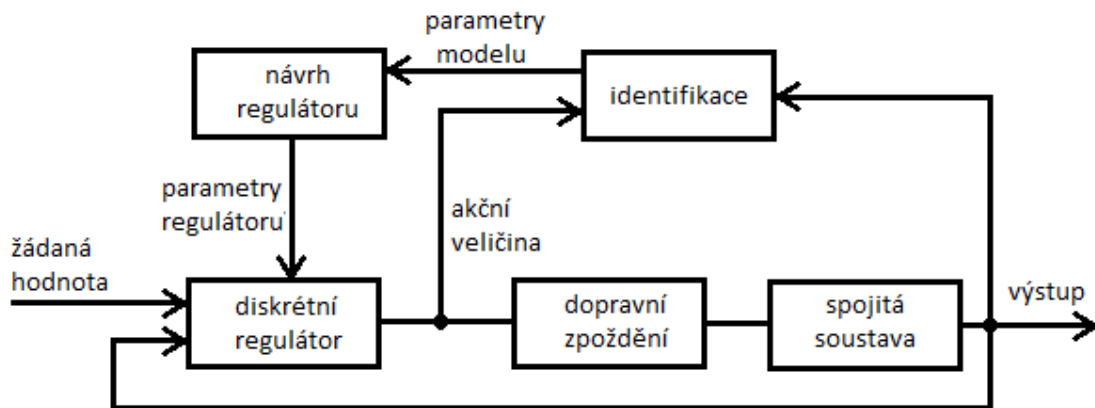
$$F_R(z) = K \left( 1 + \frac{T_s}{T_I} \frac{z^{-1}}{1-z^{-1}} + \frac{T_D}{T_s} (1-z^{-1}) \right) \quad (7)$$



Obrázek 7: Stavový diagram polohového PSD regulátoru

Symbol  $z^{-1}$  znamená zpoždění vstupního signálu o jeden krok.[ 4 ]

## 6. ŘEŠENÍ OBVODU



Obrázek 8: Teoretické blokové schéma adaptivní regulace s diskretním regulátorem

### 6.1 ZADÁNÍ

Zadání bakalářské práce má toto znění: „Simulátor adaptivních regulačních obvodů s bázi pravidel“ tzn. Vytvořit simulační program pro testování adaptivních regulačních obvodu typu STURE s bázi pravidel.

Co je regulační obvod typu STURE ( STR ) je popsáno na začátku práce.

### 6.2 BÁZE PRAVIDEL

Báze pravidel to je soupis určitých předpokladů, které jsou nutné pro správně zvolené řešení. A soupis pravidel, podle kterých se postupuje a je nutno dodržovat z důvodu správné realizovatelnosti a dodržení podmínek, které se stanovili podle metody návrhu.

#### Pravidla:

- Soustava, která bude regulována, musí být druhého řádu. Avšak její čítenel bude roven jedné. A nesmí obsahovat astatismus. Tyto

podmínky budou dále v textu zdůvodněny. Soustava bude v tomto tvaru:

$$F_S(p) = K_S \frac{1}{\alpha_2 p^2 + \alpha_1 p + \alpha_0} \quad (8)$$

$K_S$  zesílení soustavy

$\alpha_2, \alpha_1, \alpha_0$  koeficienty jmenovatele soustavy

- Identifikace je navržena tak, aby vracela model druhého řádu. Z tohoto modelu jsou použity kořeny jmenovatele, které budou tvořit časové konstanty regulátoru. Z modelu je získáno i zesílení soustavy, které má vliv na celkové zesílení otevřené smyčky.
- Regulátor je ve tvaru, kde časové konstanty jsou obsaženy v čitateli. Tento přenos může obsahovat pouze reálné časové konstanty. Jeho přenos je v tomto tvaru:

$$F_R(p) = k_R \frac{(T_1 + 1)(T_2 p + 1)}{p} \quad (9)$$

$K_R$  je zesílení regulátoru

$T_1, T_2$  jsou konstanty regulátoru

Z tohoto přenosu je zřejmé, že takový to regulátor je nerealizovatelný (z důvodu vyššího řádu čitatele než jmenovatele), ale jelikož bude výsledný regulátor diskrétní ekvivalent spojitého regulátoru, nemusíme přidávat realizační konstantu, která by zaručila realizovatelnost.

- Z předchozích dvou pravidel vyplívá následující. Je to pravidlo o tvaru přenosu otevřeného smyčky, který je dán součinem přenosu regulátoru a přenosu soustavy s dopravním zpožděním, které se přidává, pokud bude ze spojitého regulátoru přepočítávat jeho diskrétní ekvivalent. Hodnota dopravního zpoždění je polovina periody vzorkování.

$$F_R(p) = k_R \frac{(T_1 + 1)(T_2 p + 1)}{p} \quad (10)$$

$$F_S(p) = K_S \frac{1}{\alpha_2 p^2 + \alpha_1 p + \alpha_0} = \frac{K_S}{(T_1 p + 1)(T_2 p + 1)} \quad (11)$$

$$F_{Sdop}(p) = \frac{K_S}{(T_1 p + 1)(T_2 p + 1)} \cdot e^{-\frac{T_S}{2} p} \quad (12)$$

kde  $T_S$  je perioda vzorkování

$$F_O(p) = F_R(p) \cdot F_{Sdop}(p) \quad (13)$$

$$F_O(p) = k_R \frac{(T_1 + 1)(T_2 p + 1)}{p} \cdot \frac{K_S}{(T_1 p + 1)(T_2 p + 1)} \cdot e^{-\frac{T_S}{2} p} \quad (14)$$

$$F_O(p) = \frac{k_R \cdot K_S}{p} \cdot e^{-\frac{T_S}{2} p} = \frac{K_0}{p} \cdot e^{-\frac{T_S}{2} p} \quad (15)$$



Z výpočtu vyplývá, že časové konstanty v soustavě a regulátoru se vykrátí a zůstane regulátor typu I a dopravní zpoždění o polovině periody vzorkování. Toto je podstata návrhu.

- Poslední pravidlo je, aby byla zaručena fázová bezpečnost otevřené smyčky o velikosti  $60^\circ$ . Toto pravidlo je důležité pro navržení zesílení otevřené smyčky  $K_0$ .

$$F_O(p) = \frac{K_0}{p} \cdot e^{-\frac{T_s p}{2}} \quad (16)$$

Pokud je perioda vzorkování  $T_s = 0.1s$ , tak fázová charakteristika je ve tvaru:

$$\varphi(\omega) = -\frac{\pi}{2} - \frac{0.1}{2} \omega_f = -\frac{\pi}{2} - 0.05\omega_f \quad (17)$$

Fázová charakteristika v omeze řezu se musí rovnat fázové bezpečnosti  $60^\circ$ . Jelikož se fázová bezpečnost přičítá k  $-180^\circ$ , tak v bodě omega řezu musí fázová charakteristika probíhat ve fázi  $-120^\circ$ , což je požadovaná bezpečnost  $60^\circ$ . Odpovídá tomu následující rovnice:

$$\varphi(\omega) = -\frac{\pi}{2} - 0.05\omega_f = -\frac{2}{3}\pi \quad (18)$$

Z rovnice je nutné získat omega řezu:

$$-\frac{\pi}{2} - 0.05\omega_f = -\frac{2}{3}\pi \quad (19)$$

$$-0.05\omega_f = \frac{\pi}{2} - \frac{2}{3}\pi \quad (20)$$

$$\omega_f = \left(-\frac{\pi}{2} + \frac{2}{3}\pi\right) \cdot 20 \quad (21)$$

$$\omega_f = 10.47 \quad (22)$$

Ze zjištění, kde se vyskytuje omega řezu je možné pomocí amplitudové charakteristiky zjistit zesílení otevřené smyčky  $K_O$ :

$$|F_O(j\omega)| = \frac{K_O}{\omega_f} = 1 \quad (23)$$

$$K_O = \omega_f = 10.47 \quad (24)$$

Výsledný tvar přenosu otevřené smyčky je ve tvaru:

$$F_O(p) = \frac{10.47}{p} \cdot e^{-0.05p} \quad (25)$$

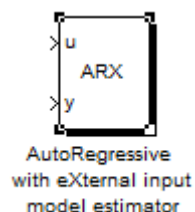
Z výsledného přenosu otevřené smyčky je zřejmé, že pro získání zesílení regulátoru  $k_R$ , je nezbytně nutné znát zesílení soustavy  $K_S$ .

$$k_R = \frac{K_O}{K_S} \quad (26)$$

### 6.3 IDENTIFIKACE

Jak již bylo řečeno identifikace je v adaptivní regulaci velice důležitá. Bez správného modelu soustavy je realizace dosti nepřesná až špatná. U adaptivní regulace je velice důležité, aby identifikace probíhala i během procesu (on-line) a ne jen jednorázově.

Program MATLAB nabízí více variant, mezi kterými bylo potřeba vybrat. Pro řešení této bakalářské práce byla vybrána identifikace metodou nejmenších čtverců, neboli ARX. V knihovně programové nadstavby MATLAB, tedy SIMULINK, je identifikační blok ARX. Ten však nebyl možný použit, z důvodu výpisu diskrétního model do workspace, až po ukončení pracovní doby. Tedy během procesu se s modelem soustavy pracovat nedalo, což bylo zásadní. Také tento blok vykresluje průběhy, které k realizaci nejsou potřebné.



**Obrázek 9: Blok ARX z knihovny SIMULINK**

Proto bylo použito pouze s-funkce (sfunarx2) z bloku ARX. Aby bylo možno použití sfunkce bylo zapotřebí vytvořit inicializační m-file (inicializace.m), ve kterém probíhá inicializace potřebných parametrů:

|                  |   |
|------------------|---|
| na=2;            | Počet časových Konstant jmenovatele                         |
| nb=2;            | Počet časových Konstant čitatele                            |
| order=[na nb 1]; | Řád polynomu $F=B/A$ [na nb nk] na...rad polynomu A         |
| npts=200;        | Velikost datového okna                                      |
| HowOften=5;      | Jak často se bude měnit model (počet vzorku)                |
| offset=0;        | Zpoždění identifikace                                       |
| ts=0.1;          | Vzorkovací perioda identifikace                             |
| method='arx';    | Metoda identifikace 'ar', 'arx', 'oe', 'armax', 'bj', 'pem' |
| mn="";           | Název proměnné, do které se bude ukládat                    |
| ph=1;            | Simulace:'inf', počet kroků dopředu:1,5,10                  |

Tyto proměnné se uloží do workspace, odkud si je již s-funkce načte, protože tyto proměnné jsou jí předávány. S těmito parametry je s-funkce (sfunarx2) funkční a identifikace probíhá.

V této fázi se docílilo pouze toho, že nebyl použit blok ARX, nýbrž jeho sfunkce, ale vlastnosti byly stejné jako samotného bloku. Byly nutné úpravy sfunkce. Úpravy spočívaly v nevýhodách samotného bloku, které byly zmíněny a to, že byly vymazány vykreslování průběhů, které byly nedůležité a hlavně k získání modelu během pracovního procesu.

Získávání modelu během pracovního procesu bylo složité v tom, že identifikace probíhá v bloku s-funkce mdlUpdate a z této části nelze posílat nic na výstup sfunkce. Řešení spočívalo v rozšíření stavového vektoru o tolik proměnných, kolik jich bylo potřeba dostat do bloku mdlOutputs. Blok sfunkce mdlOutputs slouží pro matematické operace a výstup sfunkce.

Identifikace vrací model v diskrétní formě, a protože v návrhu se používá spojitý regulátor, který bude následně přepočítán na svůj diskrétní ekvivalent, bylo zapotřebí převést diskrétní model na spojitý. Bylo také nutné ošetřit, pokud se

v čitateli soustavy objeví dostatečně malé hodnoty u jednotlivých mocnin, tak tyto hodnoty nahradit nulou, protože se jedná o nepřesnosti při identifikaci a při převodu z diskrétního na spojitý průběh. Z tohoto modelu bylo nutné získat kořeny jmenovatele a zesílení soustavy. Jelikož tyto operace jsou prováděny `mdlUpdate`, tak tyto parametry soustavy byli přidány do stavového vektoru. A `mdlOutputs` se ze stavového vektoru vyčetli a poslali se na výstup sfunkce.

#### 6.4 PŘEPOČET REGULÁTORU

Jak již bylo řečeno, regulátor je vytvořen tak, že jsou získány kořeny jmenovatele soustavy, které tvoří dvě časové konstanty v čitateli přenosu regulátoru. Tímto způsobem se docílí značného zjednodušení. Přenos regulátoru odpovídá následujícímu vztahu:

$$F_R(p) = k_R \frac{(T_1 + 1)(T_2 p + 1)}{p} \quad (27)$$

Avšak tento způsob vyjádření není vhodný na přepočtení na diskrétní ekvivalent. Z tohoto důvodu rovnici regulátoru převedeme do tvaru, který obsahuje integrační konstantu  $T_I$  a derivační konstantu  $T_D$ . Tvar přenosu regulátoru, ve kterém jsou obsaženy tyto konstanty je následující:

$$F_R(p) = K_R \left( 1 + \frac{1}{T_I p} + T_D p \right) \quad (28)$$

Abychom dostali tento tvar regulátoru, vychází se z rovnice 27, která se upravuje, až je získán tvar rovnice 28. Odvozením je docíleno přepočtených vztahů mezi jednotlivými druhy přenosů:

$$F_R(p) = \frac{k_R \cdot (T_1 + 1)(T_2 p + 1)}{p} = \frac{k_R \cdot (T_1 \cdot T_2 p^2 + (T_1 + T_2)p + 1)}{p} \quad (29)$$

$$F_R(p) = k_R \cdot \left( T_1 \cdot T_2 p + (T_1 + T_2) + \frac{1}{p} \right) \quad (30)$$

$$F_R(p) = k_R (T_1 + T_2) \cdot \left( 1 + \frac{T_1 \cdot T_2}{T_1 + T_2} p + \frac{1}{(T_1 + T_2)p} \right) \quad (31)$$

Tento přenos přímo ukazuje, jakým způsobem se počítají časové konstanty  $T_I$  a  $T_D$  a zesílení  $K_R$  v tomto tvaru.

$$K_R = k_R (T_1 + T_2) \quad (32)$$

$$T_D = \frac{T_1 \cdot T_2}{T_1 + T_2} \quad (33)$$

$$T_I = T_1 + T_2 \quad (34)$$

Z rovnice regulátoru číslo 28, již jde vytvořit diskrétní ekvivalent spojitého regulátoru. Tvar diskrétní rovnice je následující:

$$F_R(z) = K \left( 1 + d_s \frac{z^{-1}}{1 - z^{-1}} + d_d (1 - z^{-1}) \right) \quad (35)$$

Kde  $d_s$  je sumační člen (ekvivalent integrační části spojitého regulátoru) a  $d_d$  je diferenční člen diskrétního regulátoru (ekvivalent derivační části spojitého regulátoru). Přepočítání ze spojité části do diskrétní odpovídá následujícím rovnicím.

$$K = K_R \quad (36)$$

$$d_s = \frac{T_S}{T_I} \quad (37)$$

$$d_s = \frac{T_D}{T_S} \quad (38)$$

Nesmí se zapomenout, pokud se tvoří diskrétní ekvivalent ze spojitého regulátoru, tak spojitého regulátor musí být navrhován na soustavu, která je doplněna o dopravní zpoždění. Odpovídá tomu rovnice číslo 12. A až se navrhne spojitého regulátor na takto upravenou soustavu, lze vypočítat diskrétní ekvivalent spojitého regulátoru.

Z bloku, který navrhuje parametry PSD regulátoru, jsou posílány do regulátoru jeho parametry. Jedná se o integrační konstantu  $T_I$  a derivační konstantu  $T_D$ . A také zesílení soustavy, které poslouží k výpočtu zesílení regulátoru ze zesílení otevřené smyčky.

## 6.5 REALIZACE PSD REGULÁTORU

Jak bylo popsáno v předchozí kapitole, do regulátoru vstupují tři parametry z návrhu regulátoru a dva signály z procesu. Jedná se o signál žádané hodnoty  $w(t)$  a výstupní hodnoty  $y(t)$ .

Samotná realizace PSD regulátoru vychází z rovnice akčního zásahu diskrétního regulátoru:

$$u(k) = K \left( e(k) + \frac{T_S}{T_I} \sum_{i=1}^k e(i) + \frac{T_D}{T_S} (e(k) - e(k-1)) \right) \quad (39)$$

$u(k)$  je akční zásah z regulátoru

$e(k)$  je regulační odchylka  $e(k) = w(k) - y(k)$

Regulátor pokud dosud neproběhla identifikace je tvořen pouze regulační odchylkou z důvodu neznámých parametrů regulátoru a aby neovlivňoval proces identifikace.

Jakmile identifikace proběhne, tak blok regulátoru přestane posílat jako akční zásah regulační odchylku a pomocí následujících rovnic začne počítat správný akční zásah.

Vlastní program regulátoru:

- Výpočet zesílení regulátor jako podíl zesílení otevřené smyčky a zesílení modelu soustavy.

$$K = \frac{K_o}{K_s} = \frac{10.47}{K_s} \quad (40)$$

- Výpočet akčního zásahu

$$u = K \cdot (w - y) + sum + \left( \frac{K \cdot T_D}{T_S} \cdot (w - y) - ds \right) \quad (41)$$

v prvním kroku je hodnota  $ds$ ,  $sum=0$ , hodnota těchto proměnných nabývá až v druhém kroku.



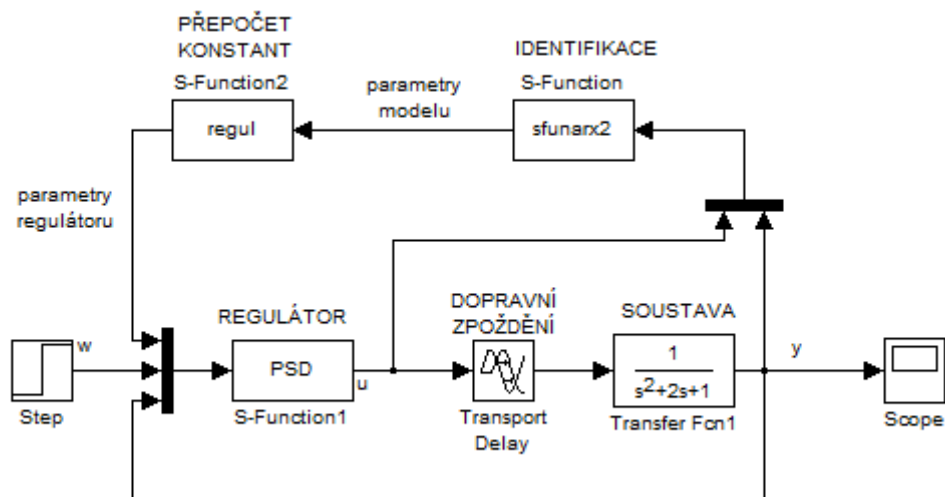
- Výpočet diference. Hodnota  $ds$  je hodnota v minulém kroku.

$$ds = \frac{K \cdot T_D}{T_s} \cdot (w - y) \quad (42)$$

- Výpočet sumace. Počáteční hodnota sumace je nulová a postupně narůstá.

$$sum = sum + \frac{K \cdot T_s}{T_I} \cdot (w - y) \quad (43)$$

## 6.6 PŘÍKLADY

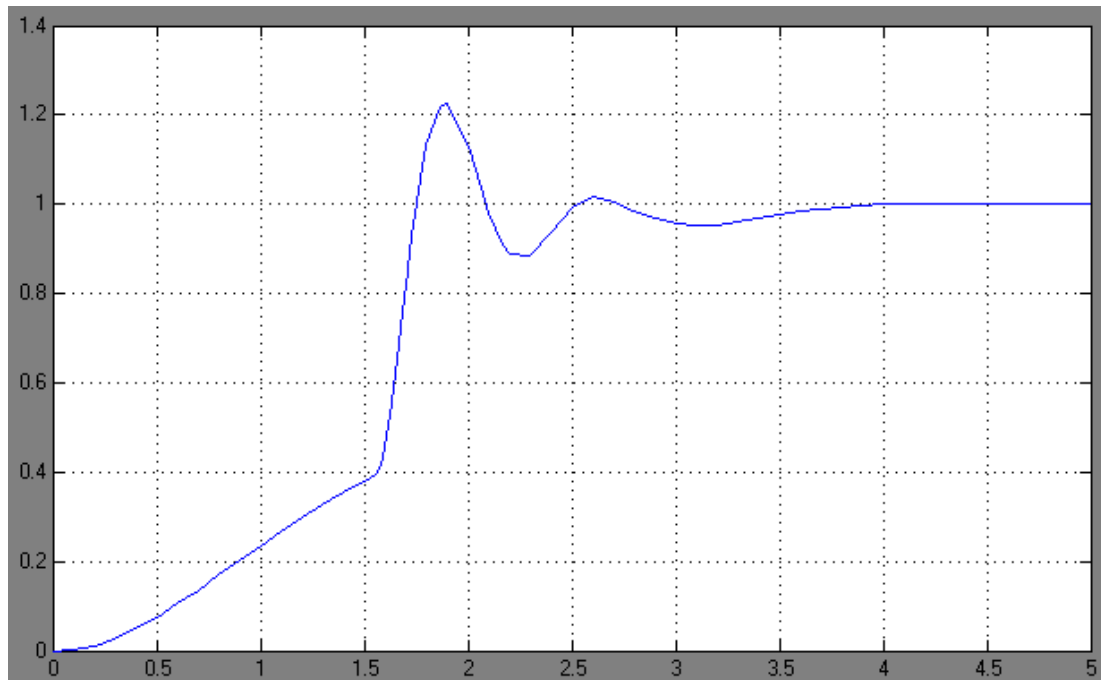


Obrázek 10: Blokové schéma reálného procesu

### 1. Soustava

$$F_s(p) = \frac{1}{p^2 + 2p + 1}$$

Jedná se o soustavu, která má dvojnásobný kořen jmenovatele o velikosti 1. A zesílení soustavy  $K_s=1$ .

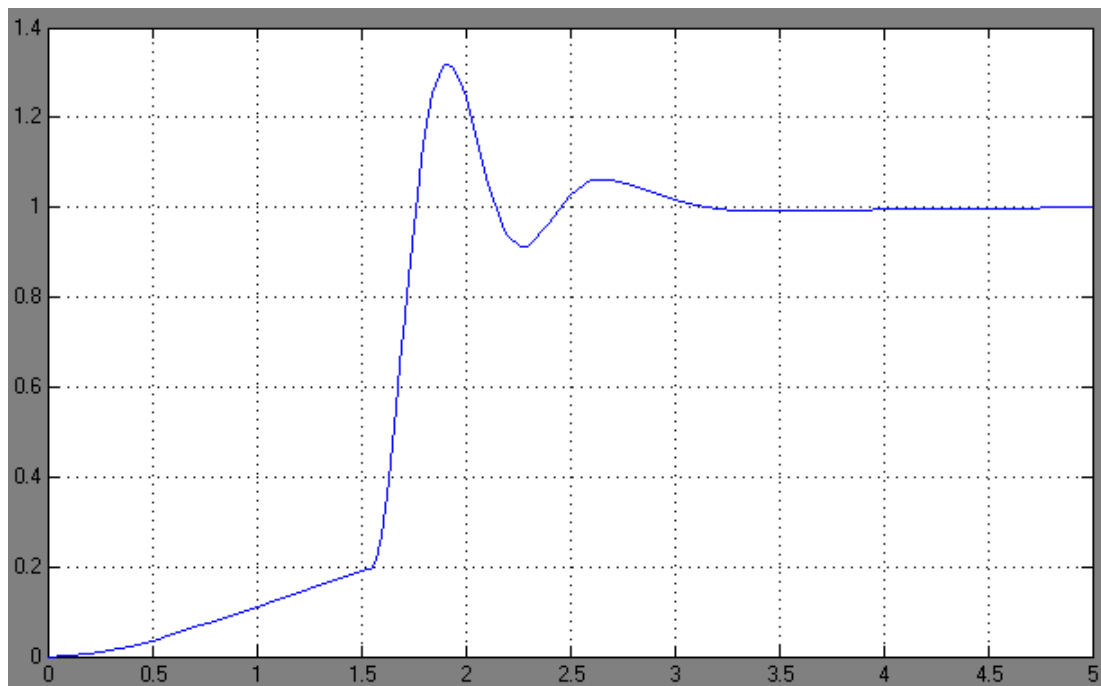


Obrázek 11: Přejchodová charakteristika pro příklad 1

## 2. Soustava

$$F_s(p) = \frac{3}{6p^2 + 15p + 3}$$

Jedná se o soustavu, která má dva různé reálné kořeny jmenovatele. Kořeny jsou -2.3 a -0.2. A zesílení soustavy  $K_s=3$



Obrázek 12: Přejchodová charakteristika pro příklad 2

## 7. ZÁVĚR

Celý simulátor adaptivního řízení je vytvořen v prostředí MATLAB a nadstavby SIMULINK.

Simulátor pracuje na principu zjednodušení otevřené smyčky obvodu, tedy regulátor kompenzuje kořeny soustavy nulami v regulátoru. Otevřená smyčka je poté vyjádřena, jako regulátor typu I a dopravní zpoždění o polovině periody vzorkování. Zesílení regulátoru se počítá na základě 60 stupňové fázové bezpečnosti otevřeného obvodu, která byla zvolena. Regulátor je typu PSD, který se počítá, jako ekvivalent PID regulátoru.

Na přechodových charakteristikách obvodů je vidět, v jaké části probíhá identifikace, a v které části již je vypočítán regulátor z modelu soustavy a probíhá regulace.

## LITERATURA

- [ 1 ] <http://www.humusoft.cz/produkty/matlab/>
- [ 2 ] °ASTRÖM,K.J.-WITTENMARK,B.AdaptiveControl.  
2.vyd.London:PrenticeHall,1994.574s.
- [ 3 ] Blaha, P., Vavřín, P.: Řízení a regulace, ET VUT Brno, 2009
- [ 4 ] Pivoňka, P.:Číslicová řídicí technika
- [ 5 ] [http://cs.wikipedia.org/wiki/Regul%C3%A1tor\\_%28automatizace%29#Literatura](http://cs.wikipedia.org/wiki/Regul%C3%A1tor_%28automatizace%29#Literatura)